This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

Freeform Search

Database: US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term: L14 and (schedule$ or assign$)

Display: 10 **Documents in Display Format:** KWIC **Starting with Number** 1

Generate: ○ **Hit List** ◉ **Hit Count** ○ **Side by Side** ○ **Image**

Search Clear Interrupt

Search History

DATE: Wednesday, September 22, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=ADJ		
L15	L14 and (schedule$ or assign$)	1	L15
L14	4709347.pn.	1	L14
L13	L12 and (network$ or packet$)	2	L13
L12	L11 and (Programmable with microengine$)	2	L12
L11	712/$.ccls. or 709/$.ccls.	24827	L11
L10	L6 and (network or packet$)	0	L10
L9	L6 and programmable	0	L9
L8	L6 and ((schedule$ or assign$) same (task$ or thread$))	1	L8
L7	L6 and ((schedule$ or assign$) and (task$ or thread$))	1	L7
L6	6463527.pn.	1	L6
L5	L1 and ((schedule$ or assign$) and (task$ or thread$))	1	L5
L4	L1 and ((schedule$ or assign$) same (task$ or thread$))	0	L4
L3	L1 and ((schedule$ or assign$) with (task$ or thread$))	0	L3
L2	L1 and (schedule$ or assign$)	1	L2
L1	6366998.pn.	1	L1

END OF SEARCH HISTORY

h e b b g e e e e e f ff e ch

Previous Doc Next Doc Go to Doc#
First Hit Fwd Refs

Generate Collection

L13: Entry 2 of 2 File: USPT Nov 24, 1987

DOCUMENT-IDENTIFIER: US 4709347 A
TITLE: Method and apparatus for synchronizing the timing subsystems of the physical modules of a local area network

Abstract Text (1):
Method and apparatus for synchronizing to a desired degree of accuracy the timing subsystems of the modules of a distributed local area network by the master and the slave. Each module includes a Module Central Processing Unit (MCPU) and a source of clock signals. Each MCPU includes a digital timing subsystem which produces a fine timing, a synchronization, and a real time timing signal. Two of the timing subsystems are provided with driver circuits one designated as the master and the other as the slave. Each timing subsystem, alternately receives the timing frames transmitted over the two cables of the network by the master and the slave. All timing subsystems other than the master, synchronize with the master. The slave transmits its timing frame in synchronization with the master.

Brief Summary Text (3):
This invention is in the field of local area networks in which a plurality of physical modules of the network communicate with one another over a network bus, and more particularly relates to methods and apparatus of synchronizing the timing subsystems of each of the modules so that they are snychronized within a predetermined degree of accuracy.

Brief Summary Text (5):
A computerized plan management system is described and claimed in co-pending Application No. 06/540,061 filed Oct. 7, 1983, now U.S. Pat. No. 4,607,256 entitled PLANT MANAGEMENT SYSTEM by Russell A. Henzel, which application is assigned to Honeywell Inc., the assignee of this application. The disclosure of Application No. 06/540,061 is hereby incorporated by reference into this application. Such a system is composed of a plurality of physical modules having varying capabilities and functionalities which communicate with one another over a common communication medium, or local control bus, to form a token-passing local area network. Each of the physical modules of the network is the equal, or peer, of the other, and each of the modules includes at least a module central processor unit (MCPU), and a module memory unit (MMU). Additional controllers and devices are added to a physical module to provide it with the ability to perform desired functions. A network of this type provides a distributed data processing environment with a concomitant increase in reliability over centralized systems since if one module fails, the network as a whole is not disabled as would be the case with a failure of a centralized system. Reliability is also improved by permitting redundancy of the physical modules of the network to the extent necessary to achieve desired system availability. Such a token-passing local control network consisting of a plurality of different types of physical modules also permits functional capabilities to be added or deleted incrementally.

Brief Summary Text (6):
One of the requirements for a computerized plant management system is that of timing the occurrence of events with a high degree of precision. A centralized timing system which could be used to satisfy the timing requirements of such a

h e b b g e e e f e e e g

plant management system would do so to the detriment of the systems' objectives of improved reliability through redundancy at the module level, of minimizing the cost of the system, and of providing additional capabilities, or modifications, to the network through the addition and deletion of physical modules since a centralized timing subsystem could not readily satisfy these objectives.

Brief Summary Text (8):
The present invention provides both method and apparatus for synchronizing to a desired degree of accuracy the timing subsystems with which each module central processor unit (MCPU) of each physical module of a local area network is provided. In such a network each physical module is the peer of every other physical module, and the physical modules of a network communicate with each other over a common, dually redundant, communication medium, i.e., two coaxial cables. Each physical module which is connected to the cables of the communication medium has the capability of transmitting bit serially binary data over these two cables at a relatively high bit rate and of receiving such signals transmitted by another module. The MCPU of each physical module includes a source of clock signals at substantially the same frequency.

Brief Summary Text (10):
The timing subsystems of at least two of the MCPU's of the network are provided with a driver circuit which when the timing subsystems is enabled by its associated MCPU permits the timing subsystems to periodically transmit timing information. Timing information is included in a timing frame which timing frames are transmitted over one of the two cables of the systems communication medium. One of the two timing subsystems equipped with a driver circuit is designated as the master and the other as the slave. The master timing subsystem transmits over one of the two cables and the slave over the other. The master and slave when the system is operating properly periodically transmit a timing frame, a set of 12 characters in the preferred embodiment, the bits of which are transmitted over the cables of the bus bit serially at a relatively low frequency. The frequency at which timing frames are transmitted is chosen so as not to interfere with the transmission of the higher bit rate signals which are also transmitted over the cables of the bus by the physical module. Each timing frame includes a synchronization code, a special set of binary signals; information as to the number of synchronization timing signals that have occurred, or have been produced since the previous one second rollover, or mark; the current, or real, time in seconds; and status information.

Brief Summary Text (11):
A timing frame is transmitted by the master and slave timing subsystems for each synchronization timing signal produced by it. The receipt of a synchronization code of a timing frame by a physical modules connected to the local control network bus is timed to coincide with the production of a synchronization timing signal by the timing subsystem, master or slave. Each timing subsystem, including the master and the slave, alternately receive timing frames from each of the two cables of the communication medium. Each timing subsystem other than the master can be commanded to synchronize its production of its synchronization timing signals with the receipt of the synchronizing code of each timing frame received from the cables of the LCN bus, and to synchronize its current real time with that of received timing frames.

Brief Summary Text (13):
It is therefore an object of this invention to provide improved methods and apparatus for synchronizing the timing subsystems of the MCPU's of each physical module of a local area network plant control system.

Brief Summary Text (14):
It is another object of this invention to provide method and apparatus for providing timing information to each module of a distributed local control network

with the required degree of accuracy, with the desired degree of reliability through redundancy, and at minimum cost.

Brief Summary Text (15):
It is yet another object of this invention to provide method and apparatus for synchronizing the timing subsystems of the modules of a local area plant control network in which synchronization signals are transmitted by designated modules over the same cables that are used to transmit all other types of information between the modules but at a different non-conflicting frequency.

Brief Summary Text (16):
It is still another object of this invention to provide a distributed timing subsystem for a plant control network in which each of the timing subsystems can be synchronized with the frequency of the source of A.C. electric power for the system.

Drawing Description Text (3):
FIG. 1 is a schematic block diagram of a local area control network.

Drawing Description Text (4):
FIG. 2 is a block diagram of the relevant portions of a physical module of a local area plant control network.

Drawing Description Text (10):
FIG. 8 illustrates the wave forms used to transmit timing information over the local control network cables.

Drawing Description Text (11):
FIGS. 9A and 9B illustrate the internal registers and the informational content of these registers utilized by each of the timing system of the MCPU of each physical module of a local area network.

Detailed Description Text (2):
The architecture of local area network 10 in which the method of this invention is practiced and the apparatus of this invention is incorporated is illustrated in FIG. 1. Physical modules 12-00 to 12-2.sup.n, where n is an integer greater than one, communicate with each other over local control network (LCN) bus 14. In network 10, each of the modules 12 is the equivalent, or the peer, of the others, and all modules 12 receive all signals transmitted over bus 14 by any of the other modules.

Detailed Description Text (3):
Each module 12, such as module 12-04 which is illustrated in FIG. 2 includes a bus interface unit (BIU) 16-04 and a transceiver 18-04 which connects BIU 16-04 to dually redundant LCN buses 14-A and 14-B. BIU 16-04 is capable of transmitting binary data over buses 14-A and 14-B and of receiving data from buses 14-A and 14-B. Transceiver 18-04, in the preferred embodiment, is transformer coupled to each of the buses 14-A and 14-B. In the preferred embodiment each of the buses 14-A and 14-B is a coaxial cable with the capability of transmitting bit serially data at a five megabit/second rate. BIU 16-04 is provided with a very fast microengine 18-04. In the preferred embodiment, microengine 18-04 is made up of bit slice components so that it can process eight bits in parallel, and can execute a 24 bit microinstruction from its programmable read only memory (PROM) 20-04 in 200 nanoseconds.

Detailed Description Text (5):
MMU 28-04 and MCPU 32-04 communicate with each other and BIU 16-04 by means of module bus 30-04. Other functions of BIU 16-04 are described in copending application Ser. No. 06/540,062 filed Oct. 7, 1983 entitled METHOD FOR PASSING A TOKEN IN A LOCAL-AREA NETWORK, by Tony J. Kozlik, which application is assigned to

the same assignee as this invention. The disclosure of the above identified application is hereby incorporated by reference into this application.

Detailed Description Text (6):
Each and every physical module 12 of local area, or control, network 10 such as module 12-04 illustrated in FIG. 2 includes a BIU 16, a transceiver 18 a module memory unit 28, an MCPU 32, and a power supply 34, which converts either 50 or 60 H.sub.z A.C. to the necessary D.C. voltage levels utilized by the components of a module 12.

Detailed Description Text (20):
Transmit mailbox TXMBOX register 100 is a 12 byte register that holds the 12 bytes of an encoded timing frame 60, which timing frame 60 will be transmitted by a master or slave timing subsystem over the local control network buses 14A or 14B when the next 50 m sec. or 1 second timing signal, or interrupt, is produced by microprocessor 56 and interrupt generator 63.

Detailed Description Text (21):
Microprocessor 56 of timing subsystem 48 does not generate directly the code written into its TXMBOX 100. The method of producing the coded data of a timing frame 60 is to translate each nibble, four bits of data, from ETI register 98 and the least significant four bits of status register 92 by conventional table look-up techniques into a set of binary digits in which no two logical zeros occur in sequence. This technique allows the transmission of one nibble per byte of the NRZ code. The NRZ zero interval following each logical one allows the use of the trailing edge of the NRZ code as a signal which generates a negative going sinusoidal shaped pulse on the line to restore its D.C. level. The reason for this is that local control network cables 14-A and 14-B cannot support D.C. wave forms, thus simple NRZ transmissions cannot be used. This limitation is overcome by transmitting wave forms on cables 14-A, 14-B in which a logical zero corresponds to a positive going sinusoidal pulse followed by a negatively going sinusoidal pulse. A logical 1 is the absence of such a pulse pair.

Detailed Description Text (24):
Timing subsystems 48 of each physical module 12 of local control network 10 have two possible configurations. If they are designed to transmit timing frames 60 the format of each of which is illustrated in FIG. 4, they are provided with a timing subsystem driver 50. If they are intended to merely receive timing frames from LCN bus 14, then a timing subsystem driver 50 is omitted. Otherwise, all timing subsystems are the same and function as described above.

Detailed Description Text (59):
From the foregoing it is clear that this invention provide methods and approached for synchronizing the timing subsystem of each of the modules of a local area network within a predetermined degree of accuracy and which satisfy the objects of this invention.

Current US Original Classification (1):
709/248

CLAIMS:

1. The method of providing synchronized and accurated timing in a distributed local area network, which network includes a plurality of modules which communicate with each other over a network bus, each module including a module central processing unit (MCPU); each MCPU including a source of clock signals, the periods of the clock signals of each of the sources of clock signals being substantially equal; each MCPU including a digital timing subsystem to which the clock signals of the sources of clock signals are applied, each of the timing subsystems producing a fine resolution timing signal, a synchronization timing signal, and a real time

timing signal, each such signal having a different period; each timing subsystem also maintaining its current real time, the number of fine resolution timing signals and the number of synchronization timing signals produced since the most recent real time timing signal was produced, and the number of fine resolution timing signals produced since the last synchronization timing signal was produced; said method comprising the steps of:

designating the timing subsystem of one of the MCPU's as a Master Timing Subsystem (MTS)

causing the MTS to transmit a timing frame over the network bus to all of the other MCPU's of the netowrk, each timing frame including a synchronizing code, the number of synchronization timing signals since the last real time timing signal was produced by the MTS, and the current real time;

causing said MTS to time the transmissions of each timing frame so that the synchronizing code of each timing frame is received by the other modules substantially at the same time as the next synchronization timing signal is produced by the MTS; and

causing each timing subsystem receiving a timing frame other than the master to synchronize its count of the number of fine resolution timing signals with that of the MTS, and its current real time with that of the MTS.

2. The method of claim 1 in which the network bus includes a pair of parallel coaxial cables, and further includes the step of:

designating a second timing subsystem as a slave timing subsystem:

causing the salve timing subsystem to transmit timing frames over one cable and the master timing subsystem to transmit timing frames over the other.

Previous Doc Next Doc Go to Doc#

h e b b g e e e f e e e g

United States Patent [19]

Kirk

[11] **Patent Number: 4,709,347**

[45] **Date of Patent: Nov. 24, 1987**

[54] **METHOD AND APPARATUS FOR SYNCHRONIZING THE TIMING SUBSYSTEMS OF THE PHYSICAL MODULES OF A LOCAL AREA NETWORK**

[75] Inventor: **David L. Kirk**, Phoenix, Ariz.

[73] Assignee: **Honeywell Inc.**, Phoenix, Ariz.

[21] Appl. No.: **682,645**

[22] Filed: **Dec. 17, 1984**

[51] **Int. Cl.**[4] **G06F 9/00**

[52] **U.S. Cl.** **364/900**; 340/825.5

[58] **Field of Search** 340/825.5; 370/82; 364/200, 900

[56] **References Cited**

U.S. PATENT DOCUMENTS

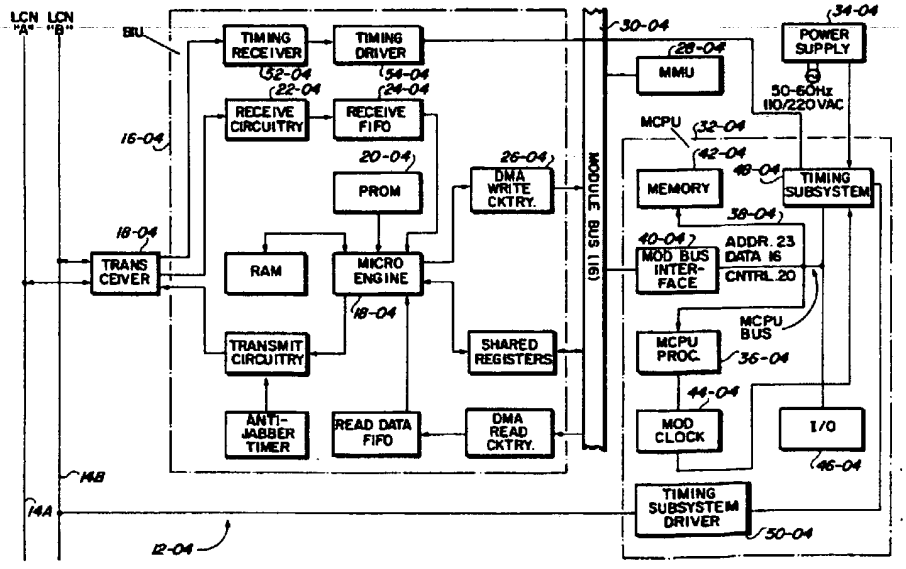
3,919,695	11/1975	Gooding	364/200
3,932,847	1/1976	Smith	364/200
4,410,889	10/1983	Bryant et al.	370/94
4,430,651	2/1984	Bryant et al.	340/825.5
4,493,021	1/1985	Agrawal et al.	364/200
4,570,162	2/1986	Boulton et al.	340/825.5

Primary Examiner—Gareth D. Shaw
Assistant Examiner—John G. Mills
Attorney, Agent, or Firm—A. A. Sapelli; A. Medved

[57] **ABSTRACT**

Method and apparatus for synchronizing to a desired degree of accuracy the timing subsystems of the modules of a distributed local area network by the master and the slave. Each module includes a Module Central Processing Unit (MCPU) and a source of clock signals. Each MCPU includes a digital timing subsystem which produces a fine timing, a synchronization, and a real time timing signal. Two of the timing subsystems are provided with driver circuits one designated as the master and the other as the slave. Each timing subsystem, alternately receives the timing frames transmitted over the two cables of the network by the master and the slave. All timing subsystems other than the master, synchronize with the master. The slave transmits its timing frame in synchronization with the master.

6 Claims, 10 Drawing Figures



Freeform Search

Database: US Pre-Grant Publication Full-Text Database; **US Patents Full-Text Database**; US OCR Full-Text Database; EPO Abstracts Database; JPO Abstracts Database; Derwent World Patents Index; IBM Technical Disclosure Bulletins

Term: L14 and (schedule$ or assign$)

Display: 10 **Documents in Display Format:** KWIC **Starting with Number** 1

Generate: ○ **Hit List** ◉ **Hit Count** ○ **Side by Side** ○ **Image**

Search | Clear | Interrupt

Search History

DATE: Wednesday, September 22, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=ADJ		
L15	L14 and (schedule$ or assign$)	1	L15
L14	4709347.pn.	1	L14
L13	L12 and (network$ or packet$)	2	L13
L12	L11 and (Programmable with microengine$)	2	L12
L11	712/$.ccls. or 709/$.ccls.	24827	L11
L10	L6 and (network or packet$)	0	L10
L9	L6 and programmable	0	L9
L8	L6 and ((schedule$ or assign$) same (task$ or thread$))	1	L8
L7	L6 and ((schedule$ or assign$) and (task$ or thread$))	1	L7
L6	6463527.pn.	1	L6
L5	L1 and ((schedule$ or assign$) and (task$ or thread$))	1	L5
L4	L1 and ((schedule$ or assign$) same (task$ or thread$))	0	L4
L3	L1 and ((schedule$ or assign$) with (task$ or thread$))	0	L3
L2	L1 and (schedule$ or assign$)	1	L2
L1	6366998.pn.	1	L1

END OF SEARCH HISTORY

h e b b g e e e e e f ff e ch

L15

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 1 of 1 returned.

☐ 1. Document ID: US 4709347 A

L15: Entry 1 of 1 File: USPT Nov 24, 1987

DOCUMENT-IDENTIFIER: US 4709347 A
TITLE: Method and apparatus for synchronizing the timing subsystems of the physical modules of a local area network

Brief Summary Text (5):
A computerized plan management system is described and claimed in co-pending Application No. 06/540,061 filed Oct. 7, 1983, now U.S. Pat. No. 4,607,256 entitled PLANT MANAGEMENT SYSTEM by Russell A. Henzel, which application is assigned to Honeywell Inc., the assignee of this application. The disclosure of Application No. 06/540,061 is hereby incorporated by reference into this application. Such a system is composed of a plurality of physical modules having varying capabilities and functionalities which communicate with one another over a common communication medium, or local control bus, to form a token-passing local area network. Each of the physical modules of the network is the equal, or peer, of the other, and each of the modules includes at least a module central processor unit (MCPU), and a module memory unit (MMU). Additional controllers and devices are added to a physical module to provide it with the ability to perform desired functions. A network of this type provides a distributed data processing environment with a concomitant increase in reliability over centralized systems since if one module fails, the network as a whole is not disabled as would be the case with a failure of a centralized system. Reliability is also improved by permitting redundancy of the physical modules of the network to the extent necessary to achieve desired system availability. Such a token-passing local control network consisting of a plurality of different types of physical modules also permits functional capabilities to be added or deleted incrementally.

Detailed Description Text (5):
MMU 28-04 and MCPU 32-04 communicate with each other and BIU 16-04 by means of module bus 30-04. Other functions of BIU 16-04 are described in copending application Ser. No. 06/540,062 filed Oct. 7, 1983 entitled METHOD FOR PASSING A TOKEN IN A LOCAL-AREA NETWORK, by Tony J. Kozlik, which application is assigned to the same assignee as this invention. The disclosure of the above identified application is hereby incorporated by reference into this application.

Detailed Description Text (39):
Certain operational tasks are scheduled to be executed by microprocessor 56 once during each 50 m sec. period. The tasks executed by microprocessor 56 depend upon the functional mode of operation of its timing subsystem 48. These unique tasks are performed at predetermined times within a 50 m sec. period. The mode of operation, local mode, clock source mode, and listener mode determine the tasks to be executed and their timing. These tasks include checking received timing frames 60 for validity, updating registers 82 and 86, checking for error limits or mode changes, processing commands for mode changes, synchronization selection changes or time

h e b b g e e e f e e ef b e

information updates. Microprocessor 56 also updates TXMBOX 100, prepares RCMBOX 94 to receive a transmitted timing frame 60 and enables interrupt generator 63 so that it can produce the appropriate interrupt each 50 m sec. and one second period. In addition microprocessor 56 updates external status register 70, determines the proper vector for the next interrupt to be produced by interrupt generator 63, and initiates the transmission of timing frames 60 at the appropriate time. Not all these tasks, however, are performed in all modes of operation of a timing subsystem 48.

Detailed Description Text (58):
Concurrently filed with this application are three applications relating to the transceiver 18, to timing subsystem driver 50, and to synchronizing the internal sense of time of a timing subsystem with the frequency of an internal A.C. power supply. These applications are entitled Dual Frequency Bus Transceiver by Robert L. Spiesman; Timing Subsystem Driver by Robert L. Spiesman; and Real Time Clock Synchronizer by David L. Kirk and Robert L. Spiesman. Each of these applications is assigned to Honeywell Inc. as in this application. The disclosue of the above identified applications are hereby incorporated by reference into this application.

Term	Documents
SCHEDULE$	0
SCHEDULE	46220
SCHEDULEABILITY	3
SCHEDULEABLE	17
SCHEDULEACKSLASH	1
SCHEDULEACTION	1
SCHEDULEACTIVELINK	1
SCHEDULEACTIVITY	1
SCHEDULEADDENTRY	1
SCHEDULEAGENDA	1
SCHEDULEALL	1
(L14 AND (SCHEDULE$ OR ASSIGN$)).USPT.	1

There are more results than shown above. Click here to view the entire set.

Display Format: KWIC Change Format

Previous Page Next Page Go to Doc#

h e b b g e e e f e e ef b e

Refine Search

Search Results -

Term	Documents
NETWORK$	0
NETWORK	279226
NETWORKA	5
NETWORKABILITY	14
NETWORKABLE	104
NETWORKACCESS	1
NETWORKACCESSIBLE	1
NETWORKADAPTOR	1
NETWORKADDRESS	11
NETWORKADDRESSES	1
NETWORKADDRESSSTRUCT	1
(L12 AND (NETWORK$ OR PACKET$)).USPT.	2

There are more results than shown above. Click here to view the entire set.

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L13

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Wednesday, September 22, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=ADJ		
L13	L12 and (network$ or packet$)	2	L13

h e b b cg b e e ch

L12	L11 and (Programmable with microengine$)	2	L12
L11	712/$.ccls. or 709/$.ccls.	24827	L11
L10	L6 and (network or packet$)	0	L10
L9	L6 and programmable	0	L9
L8	L6 and ((schedule$ or assign$) same (task$ or thread$))	1	L8
L7	L6 and ((schedule$ or assign$) and (task$ or thread$))	1	L7
L6	6463527.pn.	1	L6
L5	L1 and ((schedule$ or assign$) and (task$ or thread$))	1	L5
L4	L1 and ((schedule$ or assign$) same (task$ or thread$))	0	L4
L3	L1 and ((schedule$ or assign$) with (task$ or thread$))	0	L3
L2	L1 and (schedule$ or assign$)	1	L2
L1	6366998.pn.	1	L1

END OF SEARCH HISTORY

h e b b cg b e e ch

DOCUMENT-IDENTIFIER: US 6463527 B1
TITLE: Spawn-join instruction set architecture for providing explicit multithreading

Detailed Description Text (10):
In the example program above, Rn threads are indexed j, j+1, . . . j+Rn-1. The command assigns REGS physical registers to local virtual registers. Typically initialized to 0, global register Rb is a base register for the SUMI command of the matching Join instruction. This Spawn and Join syntax is not too different than the use of similar symbols in the high-level language "FORK," described for example in the article by C. W. Kessler and H. Seidl, "The Fork95 Parallel Programming Language: Design, Implementation, Application," International Journal on Parallel Programming, 25(1), pp. 17-50 (1997), which is incorporated herein by reference in its entirety. The assembly code also follows the style of MIPS assembly code disclosed by Patterson and Hennessy in "Computer Organization & Design. The Hardware/Software Interface," 1994, which is incorporated by reference in its entirety.

Detailed Description Text (25):
Once initiated, each TCU 34 will execute its own thread using a unique thread ID assigned to the thread being executed. Because all of the TCUs 34 will receive a set of instructions derived from a single common program, the system is referred to as a "single program multiple data (SPMD)" system. Preferably, a copy of the thread instructions (referred to as "Spawn-Join instructions") is transferred on the bus from instruction memory 33 to local memory in each TCU 34. Although the instructions retrieved into TCU local memory may be the same for each of the TCUs 34, the interpretations made by each individual TCU 34a, 34b, 34c, . . . 34k will be different based on the individual thread ID and data parameters in associated registers R1 . . . R64 of register file 30 used at the time. In the preferred embodiment, TCUs 34a, 34b, 34c, . . . 34k will be initially assigned to execute threads having thread ID numbers 1, 2, 3, . . . k, respectively. Threads corresponding to thread ID numbers k+1, k+2, . . . n will be subsequently executed by individual TCUs 34 in turn as they terminate current execution of their respective threads.

Detailed Description Text (54):
During the nesting of Spawn commands in the EMT model, the TCU assigns and stores a unique identification (ID) number to each active thread. This ID information is maintained in a table, together with spawning information regarding the relative position of each thread to predecessor ("parent") and successor ("child") threads. When a thread executes a Join instruction, the thread is terminated and control reverts back to the "parent" thread. Once all active threads have been terminated, a transition to the serial state is made, as in the operation described above.

CLAIMS:

10. A computer system for processing a parallel algorithm having a parallel code block with n virtual threads, the computer system comprising: a spawn control unit initiating execution of k physical threads by generating a thread control unit

h e b b g e e e f e e e g

enable signal in a form of a spawn command, assigning each thread a thread identification number; a plurality of thread control units, wherein each thread control unit receives the spawn command from said spawn control unit, and in response to the spawn command, retrieves a series of spawn-join instructions from a global instruction memory, each series of spawn-join instructions including a join command signaling a termination of a thread upon execution by a thread control unit, wherein said thread control units execute their respective series of spawn-join instructions in concurrently, and wherein each thread control unit executes its respective series of spawn-join instructions independent of any order of execution of spawn-join instructions by other thread control units; a prefix-sum unit, coupled to each of said thread control units, calculating a plurality of prefix sums based on outputs from said thread control units, and wherein thread identification numbers are assigned to said thread control units based on calculations of the prefix sums; wherein each of said thread control units sends an output to said prefix-sum unit in response to execution of a join command, and if the number of k physical threads is less than the number of n virtual threads, said spawn control unit issues a thread control unit enable signal in a form of a spawn-recur command when at least one of said thread control units has executed a join command, wherein each thread control unit receiving said spawn-recur command commences recurrent execution of its respective series of spawn-join instructions with a new thread identification number from said prefix-sum unit.

13. In a computer system, the method of processing a parallel algorithm having n virtual threads, the method comprising the steps of: initiating execution of k physical threads by generating a thread enable signal in a form of a spawn command and assigning each thread a thread identification number; receiving the spawn command, and in response to the spawn command, retrieving a series of spawn-join instructions, each series of spawn-join instructions including a join command signaling a termination of a thread upon execution; executing respective series of spawn-join instructions in parallel and independent of any order of execution of spawn-join instructions; calculating a plurality of prefix sums based on terminating ones of the k physical threads, and assigning thread identification numbers based on calculations of the prefix sums; and wherein, if the number of k physical threads is less than the number of n virtual threads, issuing a thread enable signal in a form of a spawn-recur command when at least one join command has been executed, wherein in response to said spawn-recur command, commencing recurrent execution of a series of spawn-join instructions with a new thread identification number output from said prefix-sum step.

Previous Doc Next Doc Go to Doc#

US006463527B1

(12) **United States Patent**
Vishkin

(10) **Patent No.: US 6,463,527 B1**
(45) **Date of Patent: Oct. 8, 2002**

(54) **SPAWN-JOIN INSTRUCTION SET ARCHITECTURE FOR PROVIDING EXPLICIT MULTITHREADING**

(75) Inventor: **Uzi Y. Vishkin**, 2 Dundee Ct., Rockville, MD (US) 20850

(73) Assignee: **Uzi Y. Vishkin**, Rockville, MD (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/380,571**

(22) PCT Filed: **Mar. 20, 1998**

(86) PCT No.: **PCT/US98/05975**

§ 371 (c)(1),
(2), (4) Date: **Sep. 7, 1999**

(87) PCT Pub. No.: **WO98/43193**

PCT Pub. Date: **Oct. 1, 1998**

Related U.S. Application Data

(60) Provisional application No. 60/041,044, filed on Mar. 21, 1997, and provisional application No. 60/071,516, filed on Jan. 15, 1998.

(51) **Int. Cl.**[7] **G06F 9/44**; G06F 9/52; G06F 9/30

(52) **U.S. Cl.** **712/245**; 712/228; 712/219; 712/242; 709/107; 709/108; 709/315

(58) **Field of Search** 709/100, 106, 709/103, 108, 223, 315, 316, 107; 707/3, 533; 712/22, 23, 30, 203, 216, 200, 215, 24, 25, 204, 205, 206, 207, 217, 223, 226, 235, 245, 208, 209, 213, 210, 211, 212, 218, 219, 228, 227, 237, 238, 21, 42, 41, 242, 243; 717/150

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,050,070	A	9/1991	Chastain et al.	712/203
5,151,991	A	9/1992	Iwasawa et al.	717/150
5,557,747	A *	9/1996	Rogers et al.	709/223
5,666,524	A *	9/1997	Kunkel et al.	707/3
5,771,382	A *	6/1998	Wang et al.	709/100
5,812,811	A	9/1998	Dubey et al.	712/216

FOREIGN PATENT DOCUMENTS

WO	WO 98/43193	* 10/1998

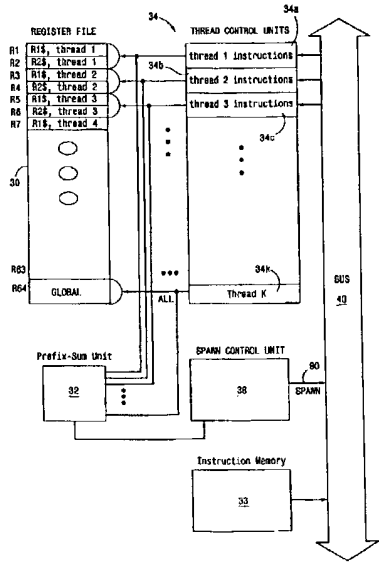
* cited by examiner

Primary Examiner—Daniel H. Pan
(74) *Attorney, Agent, or Firm*—Dickstein Shapiro Morin & Oshinsky LLP

(57) **ABSTRACT**

The invention presents a unique computational paradigm that provides the tools to take advantage of the parallelism inherent in parallel algorithms to the full spectrum from algorithms through architecture to implementation. The invention provides a new processing architecture that extends the standard instruction set of the conventional uniprocessor architecture. The architecture used to implement this new computational paradigm includes a thread control unit **(34)**, a spawn control unit **(30)**, and an enabled instruction memory **(50)**. The architecture initiates multiple threads and executes them in parallel. Control of the threads is provided such that the threads may be suspended or allowed to execute each at its own pace.

14 Claims, 5 Drawing Sheets



Refine Search

Search Results -

Term	Documents
THREAD$	0
THREAD	135225
THREADA	1
THREADAABLE	1
THREADAADDRISINTHREAD	1
THREADABELY	1
THREADABILITY	33
THREADABILY	2
THREADABIY	3
THREADABLE	5653
THREADABLELY	2
(L14 AND THREAD$).USPT.	0

There are more results than shown above. Click here to view the entire set.

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L17

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Wednesday, September 22, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=ADJ		
L17	L14 and thread$	0	L17

h e b b cg b e e ch

L16	L14 and packet$	0	L16
L15	L14 and (schedule$ or assign$)	1	L15
L14	4709347.pn.	1	L14
L13	L12 and (network$ or packet$)	2	L13
L12	L11 and (Programmable with microengine$)	2	L12
L11	712/$.ccls. or 709/$.ccls.	24827	L11
L10	L6 and (network or packet$)	0	L10
L9	L6 and programmable	0	L9
L8	L6 and ((schedule$ or assign$) same (task$ or thread$))	1	L8
L7	L6 and ((schedule$ or assign$) and (task$ or thread$))	1	L7
L6	6463527.pn.	1	L6
L5	L1 and ((schedule$ or assign$) and (task$ or thread$))	1	L5
L4	L1 and ((schedule$ or assign$) same (task$ or thread$))	0	L4
L3	L1 and ((schedule$ or assign$) with (task$ or thread$))	0	L3
L2	L1 and (schedule$ or assign$)	1	L2
L1	6366998.pn.	1	L1

END OF SEARCH HISTORY

h e b b cg b e e ch

Refine Search

Search Results -

Term	Documents
PACKET$	0
PACKET	50147
PACKETA	3
PACKETABLE	2
PACKETALWAYS	1
PACKETAND	4
PACKETARBITRATION	1
PACKETARE	1
PACKETARGS	1
PACKETARRAY	2
PACKETARRIVAL	3
(L14 AND PACKET$).USPT.	0

There are more results than shown above. Click here to view the entire set.

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L16

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Wednesday, September 22, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=ADJ		
L16	L14 and packet$	0	L16

h e b b cg b e e ch

L15	L14 and (schedule$ or assign$)	1	L15
L14	4709347.pn.	1	L14
L13	L12 and (network$ or packet$)	2	L13
L12	L11 and (Programmable with microengine$)	2	L12
L11	712/$.ccls. or 709/$.ccls.	24827	L11
L10	L6 and (network or packet$)	0	L10
L9	L6 and programmable	0	L9
L8	L6 and ((schedule$ or assign$) same (task$ or thread$))	1	L8
L7	L6 and ((schedule$ or assign$) and (task$ or thread$))	1	L7
L6	6463527.pn.	1	L6
L5	L1 and ((schedule$ or assign$) and (task$ or thread$))	1	L5
L4	L1 and ((schedule$ or assign$) same (task$ or thread$))	0	L4
L3	L1 and ((schedule$ or assign$) with (task$ or thread$))	0	L3
L2	L1 and (schedule$ or assign$)	1	L2
L1	6366998.pn.	1	L1

END OF SEARCH HISTORY

h e b b cg b e e ch

Refine Search

Search Results -

Term	Documents
SCHEDUL$	0
SCHEDUL	9
SCHEDULA	1
SCHEDULABILITY	44
SCHEDULABLE	215
SCHEDULABLE-THIS	1
SCHEDULABLY	1
SCHEDULAGE	1
SCHEDULAING	1
SCHEDULAR	70
SCHEDULAR/RESOURCE	1
(L21 AND SCHEDUL$).USPT.	1

There are more results than shown above. Click here to view the entire set.

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L22

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Wednesday, September 22, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=YES; OP=ADJ			
L22	L21 and schedul$	1	L22

h e b b cg b e e ch

L21	L1 and thread$	1	L21
L20	L18 and (schedul$ with thread$)	1	L20
L19	L18 and (schedul$ with thread$ with process$)	0	L19
L18	L1 or L6	2	L18
L17	L14 and thread$	0	L17
L16	L14 and packet$	0	L16
L15	L14 and (schedule$ or assign$)	1	L15
L14	4709347.pn.	1	L14
L13	L12 and (network$ or packet$)	2	L13
L12	L11 and (Programmable with microengine$)	2	L12
L11	712/$.ccls. or 709/$.ccls.	24827	L11
L10	L6 and (network or packet$)	0	L10
L9	L6 and programmable	0	L9
L8	L6 and ((schedule$ or assign$) same (task$ or thread$))	1	L8
L7	L6 and ((schedule$ or assign$) and (task$ or thread$))	1	L7
L6	6463527.pn.	1	L6
L5	L1 and ((schedule$ or assign$) and (task$ or thread$))	1	L5
L4	L1 and ((schedule$ or assign$) same (task$ or thread$))	0	L4
L3	L1 and ((schedule$ or assign$) with (task$ or thread$))	0	L3
L2	L1 and (schedule$ or assign$)	1	L2
L1	6366998.pn.	1	L1

END OF SEARCH HISTORY

h e b b cg b e e ch

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 1 of 1 returned.

☐ 1. Document ID: US 6366998 B1

L22: Entry 1 of 1 File: USPT Apr 2, 2002

DOCUMENT-IDENTIFIER: US 6366998 B1
**** See image for Certificate of Correction ****
TITLE: Reconfigurable functional units for implementing a hybrid VLIW-SIMD programming model

Brief Summary Text (5):
Instructions are processed by a scheduler which determines which functional units should be used for executing each instruction. Scheduling may be done statically, i.e., at compile time, as opposed to dynamically, i.e., at run time. Thus, VLIW models can simultaneously execute instructions while minimizing the occurrence of hazards. Because of this feature, among others, instruction parallelism models are very efficient in telecommunications applications.

Brief Summary Text (6):
Developing an instruction set architecture based on a VLIW model has several advantages. First, VLIW models are very scalable, both upward and downward. Scalability refers to the number of operations that can be packed into one long instruction word. The scalability enables the model to serve as a basis for a family of derivative implementations for various high performance digital signal processor ("DSP") and multimedia applications. Second, "memory walls" are not an issue in the VLIW model. Memory walls refer to the concept that processor speeds are increasing at a rate more quickly than memory speeds. In the case of a VLIW model, memory walls are not a concern because the processor is simultaneously executing a large number of instructions instead of executing one complex instruction in a consecutive order where a processor would have to repeatedly wait for information from memory for every consecutive instruction. Third, the VLIW model saves silicon area and power by off loading the complex instruction scheduling scheme to the compiler.

Brief Summary Text (13):
Accordingly, the present invention overcomes problems in the prior art by providing an instruction set architecture for a digital signal processor that has improved code density, improved instruction level parallelism and improved issue bandwidth. The instruction set architecture includes information packets which may include instructions having different characteristics, such as instruction type (for example, scalar or vector) and instruction length (for example, 16-bit and 32-bit). These instructions are received by a scheduler or scoreboard unit which then determines the functional units that are available for executing the instructions. The instructions are then broadcast to a plurality of function units or processing elements for execution.

Drawing Description Text (13):
FIG. 11 is a schematic representation of an exemplary scheduler.

h e b b g e e e f e e ef b e

Detailed Description Text (6):
Instructions 104-116 of the hybrid VLIW-SJMD DSP 100 are preferably received by a scheduler or scoreboard unit 120 which then determines which functional units are available for executing the instructions. Instructions 104-116 are then broadcast data path units ("DPUs") 122, each of which typically includes a plurality of functional units or processing elements. An exemplary DSP 100 preferably includes five DPUs 122. The functional units or processing elements included within DPUs 122 execute instructions 104-116 utilizing data element or operands from a scalar register file 124 and a vector register file 126.

Detailed Description Text (15):
Mode bits are typically located in a multiple-bit template field identifying specifications of an instruction packet which is contained in each instruction packet in accordance with a preferred embodiment of the present invention. In addition to a mode bit sub-field, a template field may also include the following sub-fields: a grouping field (which contains instruction issue groups), a thread identifier, and a repeat field (which identifies whether the entire instruction packet is to be repeated as a zero-overhead loop).

Detailed Description Text (17):
Related to the issue group is the issue bandwidth, which represents the number of simple instructions that can be issued, i.e., physically dispatched to execution units, per second. If the issue bandwidth is much smaller than the fetch bandwidth, i.e., the number of VLIW fetch packets that a DSP can fetch per second, the performance of the processor will deteriorate significantly. In other words, a DSP will not be operating efficiently if it is fetching instructions faster than it is executing the instructions and creating a buildup or backlog of instructions. This may be a result of a largely serial or scalar application, which does not take advantage of the parallel resources provided by the processor. This may also be the result of poor instruction scheduling in that the scheduler is not searching broadly enough for independent instructions that can be issued out of order and that can utilize the issue bandwidth of the processor. To achieve more efficient processing, VLIW fetch packets are preferably filled to capacity to take advantage of code density and issue groups are preferably maximized within each fetch packet.

Detailed Description Text (24):
The configurability of the present hybrid VLIW-SIMD DSP may require intelligence in the hardware to execute the operations in the instruction packets. In general, instruction packets are broadcast to a plurality of processing elements or functional units where each instruction packet contains instructions of various characteristics as discussed above. These characteristics may include, inter alia, varying instruction types and varying instruction lengths. The instruction packet need not identify which specific functional units should be used in executing the various types of instructions. Rather, a scheduler in a DSP is preferably designed to schedule instructions for particular functional units depending on the specific instructions. In a subsequent cycle, the scheduler may reconfigure the coupling or grouping of the functional units and schedule different instructions to them for execution. The reconfiguration ability reduces the amount of execution time needed and reduces the chance of hazards, such as read after write ("RAW") hazards. The functional unit configurability is preferably facilitated by buses feeding source operands to the functional units, buses transferring the results, and the scheduling logic for implementing result forwarding and bypass paths.

Detailed Description Text (32):
The hardware may determine which ~~of the functional units~~ are ~~available to perform~~ the operations specified in the instruction packets via a hardware scheduler, reconfiguring element or scoreboard unit ~~which tracks~~ which functional units are currently performing operations and which units are available to perform

h e b b g e e e f e e ef b e

operations. The scheduler determines functional unit availability in a way such that the comparators are minimized and the cycle time is not increased. With reference to FIG. 1, an exemplary scheduler 1100 is shown charting destination operands 1102 against source operands 1104. For example, if a particular ALU is currently performing an operation, scheduler 1100 identifies what other ALUs are available to receive data to perform an operation and which ALUs cannot receive the data. Scheduler 1100 is configured based upon how the DPUS and other processing elements are associated and connected.

CLAIMS:

15. The processor of claim 14, wherein said reconfiguring element is configured to receive said first instructions and said second instructions, to determine the availability of said functional units and to schedule said first instructions and said second instructions to said functional units for execution.

18. The digital signal processing system of claim 17, wherein said reconfiguring element receives said first instructions, said second instructions, and said third instructions, to said functional units and wherein said functional units execute said scheduled instruction.

Term	Documents
SCHEDUL$	0
SCHEDUL	9
SCHEDULA	1
SCHEDULABILITY	44
SCHEDULABLE	215
SCHEDULABLE-THIS	1
SCHEDULABLY	1
SCHEDULAGE	1
SCHEDULAING	1
SCHEDULAR	70
SCHEDULAR/RESOURCE	1
(L21 AND SCHEDUL$).USPT.	1

There are more results than shown above. Click here to view the entire set.

h e b b g e e e f e e ef b e